ONLINE COURSE REGISTER SYSTEM PROJECT REPORT.

Research Proposal · August 2023

DOI: 10.13140/RG.2.2.36511.78249

1 author:

Kamal Acharya
Tribhuvan University
192 PUBLICATIONS **2,699** CITATIONS

SEE PROFILE

AN

PROJECT REPORT

ON

ONLINE COURSE REGISTER SYSTEM PROJECT

BY

KAMAL ACHARYA

(Tribhuvan University)

Date: 2023/08/10

CHAPTER 1

PROJECT DESCRIPTION

1.1 INTRODUCTION

Student course registration process in colleges involve filling registration forms manually, getting it signed by respective subject teachers, and then getting the documents acknowledged from the concerned Advisors, College Deans and Accounts Officers respectively. Finally the registration forms are submitted in the Administrative Branch. As is evident, this process is very laborious and time consuming. An Online Student Course Registration System has been developed to simplify the current manual procedure. This system has been developed using PHP and MySQL. The front-end is designed using PHP with excerpts of code written using and back-end is designed and managed through MySQL. This system software is more secured, user-friendly and less time-consuming. Basically, systems are implemented for facilitating complex manual processes and that is exactly what we are trying to achieve. System is implemented as per user requirement such as a manufacturing concern may install a plant for easing out manual processes. We have sought help from computer programming for automation of manual registration system. With the introduction of computers, every aspect of our lives has been revolutionized. When used judiciously, computers can help us save time, secure our personal information, access the required information whenever and wherever required. Keeping all these positive points in mind, we have developed an Online Student Course Registration System for easily managing the semester registration process for the student in an institution. Ours is an advisory based system. In state agricultural universities the course allocation is advisory based and more complicated. The courses are assigned according to the skill set and industry requirements. Hence, in current scenario, automated system is required for course registration of students.

1.2 EXISTING SYSTEM

The existing system involves the student filling five identical forms called yellow forms which contain student's personal details as well as the courses he/she has to register during the next semester. After filling those forms, the student gets them signed by the respective subject teachers and the Advisor assigned. Then the student submits the semester fees and obtains the fee receipt and gets the signature of the Accounts Officer by obtaining the fee receipt. After this process, Dean Nominees acknowledges and confirms all the signatures and receipts and finally the forms are submitted in the administrative block. The student also receives one of the copies of the Yellow Form. As is evident, the student and faculty are hassled whilst completing the registration formalities and moreover, it leads to an added discomfort.

DISADVANTAGES

- User have to fill many forms to register their details
- Needs many signatures from the head officials for the acceptance

1.2 PROPOSED SYSTEM

The user can register here in the registration form which is developed using the web application developed. They don't need any sign of the officials since admin will verify the registered details. The student can pay the registration fees through the online transaction. Thus they can easily select the course they want to study using the online course registration web application.

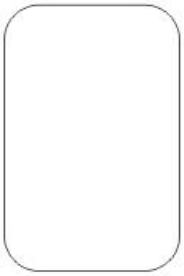
ADVANTAGES

- Eliminate all the manual intervention and increase the speed of whole process
- System will allow student to fill the form online, system has inbuilt validation system to validate the entered data
- Allows the administrator to access the students details quickly

2.1 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in.

Data flow Symbols:

Symbol	**Description**
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

LEVEL 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts

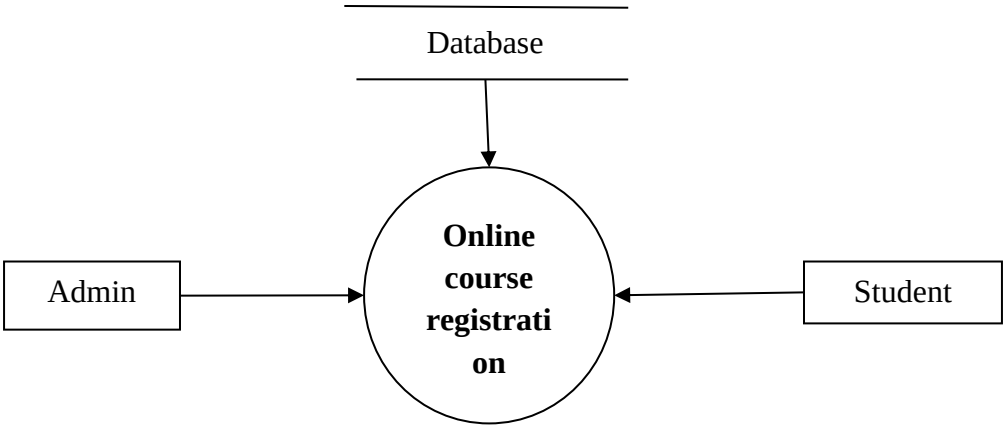


Fig 2.1.1 level 0-DFD

LEVEL 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes.

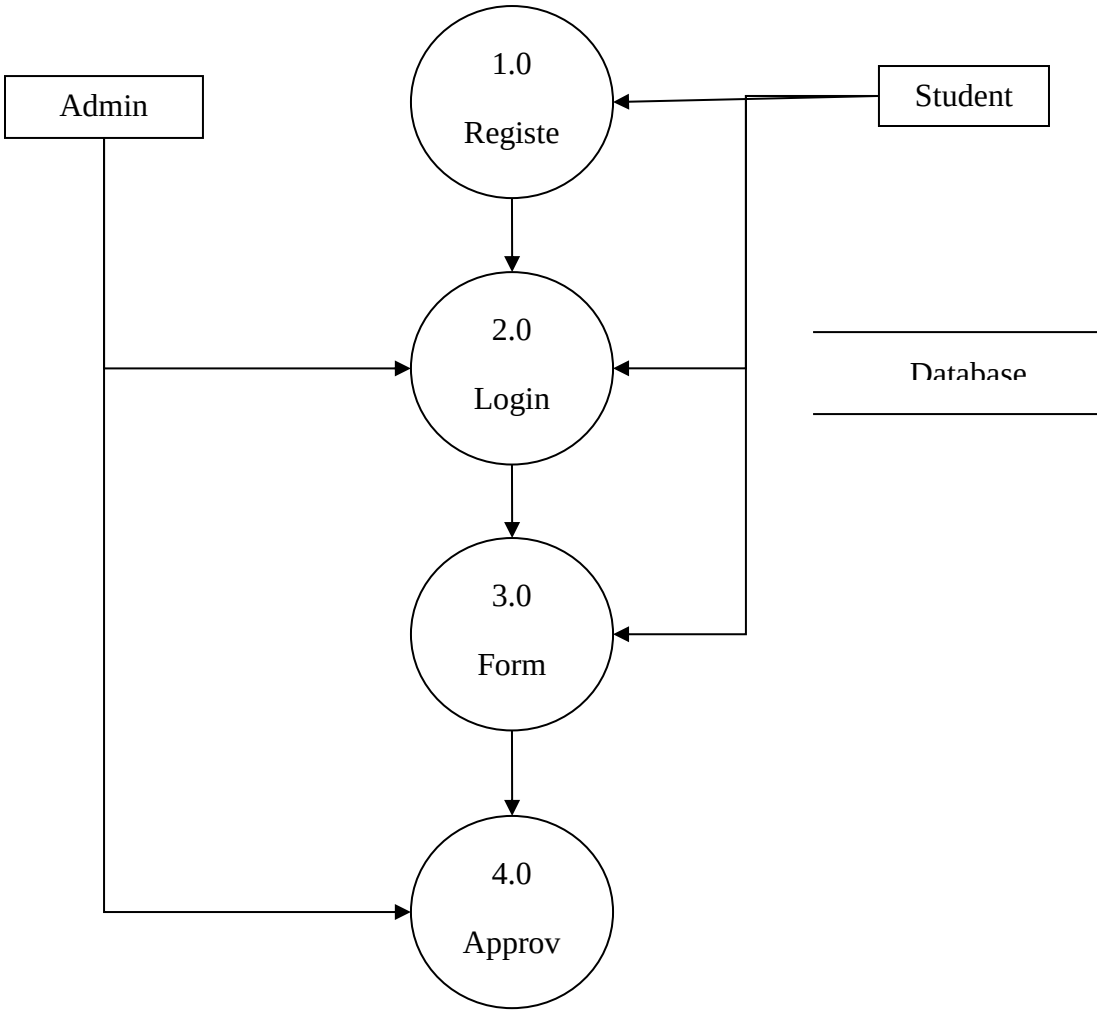


Fig 2.1.2 level 1 DFD

2.2 ARCHITECTURAL DIAGRAM

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software.

An architecture diagram is a graphical representation of a set of concepts that are part of architecture, including their principles, elements and components. Architecture diagram can help system designers and developers visualize the high-level, overall structure of their system or application, in order to ensure the system meets their users' needs. Using architecture diagram, you can also describe patterns that are used throughout the design. It's somewhat like a blueprint that you use as a guide, so that you and your colleagues can discuss, improve and follow.

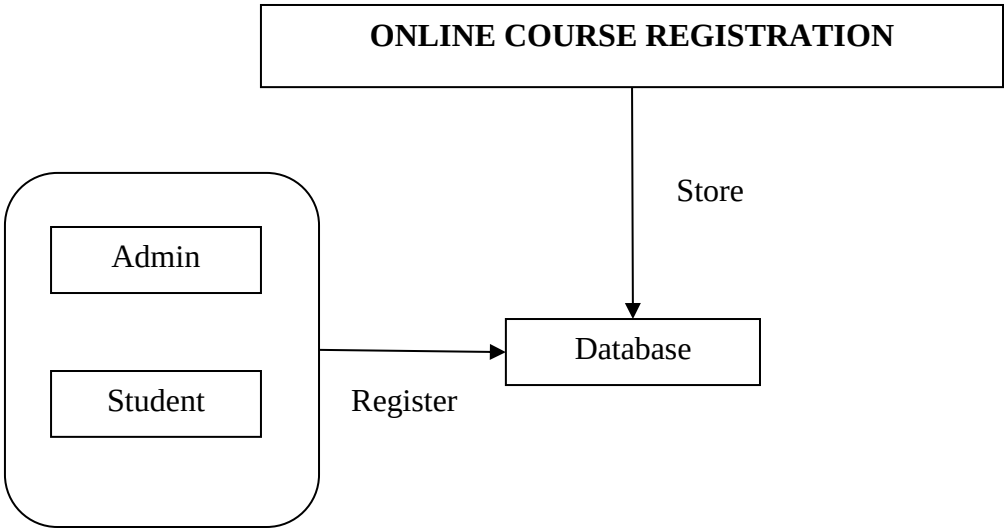


Fig 2.2.1 System Architecture

CHAPTER 3

DATABASE DESIGN

3.1 DATA DICTIONARY

3.3 RELATIONSHIP DIAGRAM

An entity–relationship model (ER model for short) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types). In software_engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data_model, that defines a data or information structure which can be implemented in a database, typically a relational_database.

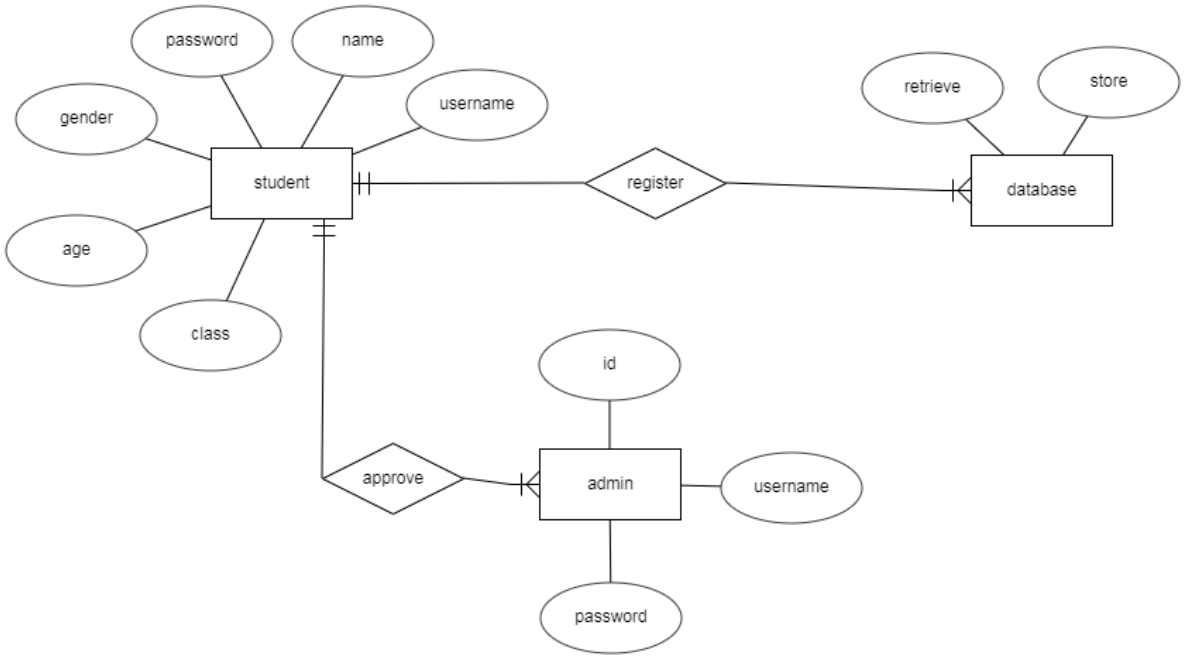


Fig 3.3.1 ER diagram

CHAPTER 4
PROGRAM DESIGN

4.1 MODULES

- Admin module
- Student module

4.2 MODULES DESCRIPTION

Admin module:

In admin module, admin will add the details of the course, description about the course, duration of the course and so on. The admin will accept the request of the student who is registered for the course.

1. **Add course:**

The course and the description about the course will be added so that the student can know about the course they wish to study. Admin will also add the duration of the course which is added.

2. **Add amount:**

The amount which is needed for the course is added with the description details. The amount for each course will be added

Student module:

In this module, student will register their details and will get a login session with a username and password.

1. **Fill form:**

The student after registration they will fill the form with the details which are required to the admin for course registration

2. **Online payment**

After registration, the student will pay the amount through the online transaction. The amount will be transferred to the admin.

CHAPTER 5
SYSTEM TESTING

5.1 SYSTEM TESTING

Testing

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system. The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

5.1.1 Types of Testing:

5.1.2 System testing

After a system has been verified, it needs to be thoroughly tested to ensure that every component of the system is performing in accordance with the specific requirements and that it is operating as it should including when the wrong functions are requested or the wrong data is introduced.

Testing measures consist of developing a set of test criteria either for the entire system or for specific hardware, software and communications components. For an important and sensitive system such as an electronic voting system, a structured system testing program may be established to ensure that all aspects of the system are thoroughly tested.

Testing measures that could be followed include:

- Applying functional tests to determine whether the test criteria have been met
- Applying qualitative assessments to determine whether the test criteria have been met.
- Conducting tests in “laboratory” conditions and conducting tests in a variety of “real life” conditions.

- Conducting tests over an extended period of time to ensure systems can perform consistently.
- Conducting “load tests”, simulating as close as possible likely conditions while using or exceeding the amounts of data that can be expected to be handled in an actual situation.

Test measures for hardware may include:

- Applying “non-operating” tests to ensure that equipment can stand up to expected levels of physical handling.
- Testing “hard wired” code in hardware (firmware) to ensure its logical correctness and that appropriate standards are followed.

Tests for software components also include:

- Testing all programs to ensure its logical correctness and that appropriate design, development and implementation standards have been followed.
- Conducting “load tests”, simulating as close as possible a variety of “real life” conditions using or exceeding the amounts of data that could be expected in an actual situation.
- Verifying that integrity of data is maintained throughout its required manipulation.

5.1.2 Unit testing

The first test in the development process is the unit test. The source_code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed.

Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

5.1.4 System testing

Several modules constitute a project. If the project is long-term project, several developers write the modules. Once all the modules are integrated, several errors may arise. The testing done at this stage is called system test. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. Testing a specific hardware/software installation. This is typically performed on a COTS (commercial off the shelf) system or any other system comprised of disparate parts where custom configurations and/or unique installations are the norm.

5.1.5 Integration testing

Testing is which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

CHAPTER 6

CONCLUSION

In this system, we described our iterative approach to designing a web-based course registration system. We started by analyzing the usability issues in course registration system and developed a prototype that overcomes these problems. We gathered feedback on our prototype and re-fined it accordingly. Finally, we performed a summative evaluation to assess the usability of our prototype and my Student System. Our summative evaluation results showed that our design was able to reduce the amount of time needed by participants to register through online , remove the need to use external tools to create a class schedule, and improve the overall user satisfaction with the system. Most participants preferred online course registration system instead of filling the form and getting sign from the higher officials. Based on our observations and analysis, we developed design guidelines for web-based course registration systems. We expect that our model can assist software developers in designing and implementing such systems at universities and help them to register and pay fees through online.

CHAPTER 7

REFERENCES

BIBLIOGRAPHY

Book reference:

[1] Alvin J Bruney M V P,"The Microsoft Office Web Components Black Book with .NET ", Lulu.com (9 April 2005)

[2] Brennan Stehling," Pro ASP.NET for SQL Server: High Performance Data Access for Web Developers (Expert's Voice in SQL Server)", Apress; 1st ed. edition (26 September 2007)

[3] Robin Dewson," Beginning SQL Server 2005 Express for Developers: From Novice to Professional (Expert's Voice in .NET) ", Apress; 1st ed. edition (8 February 2007)

[4] Matthew Macdonald, " ASP.NET: The Complete Reference", McGraw Hill Education (1 July 2017)

Web reference:

1. https://www.researchgate.net/publication
2. http://www3.cis.gsu.edu
3. https://ieeexplore.ieee.org/search

Author reference:

Acharya, Kamal. "STUDENT INFORMATION MANAGEMENT

SYSTEM." Authorea Preprints (2023).

Acharya, Kamal. "Library Management System." Available at SSRN4807104 (2019).

ACHARYA, KAMAL, et al. "LIBRARY MANAGEMENT SYSTEM." (2019).

Acharya, Kamal. "Online bus reservation system project report." Authorea

Preprints (2024).

Acharya, Kamal. "Online bus reservation system project report." (2024).

Acharya, Kamal. “Online Bus Reservation System.” SSRN ElectroNIC ASIA Journal (2024): n. pag.

Acharya, Kamal. “Student Information Management System Project.” SSRN ElectroNIC ASIA Journal (2024): n. pag.

Acharya, Kamal. “ATTENDANCE MANAGEMENT SYSTEM.” International Research Journal of Modernization in Engineering Technology and Science (2023): n. pag.

Acharya, Kamal. “College Information Management System.” SSRN ElectroNIC ASIA Journal (2024): n. pag.

Acharya, Kamal, Attendance Management System Project (April 28, 2024). Available at SSRN: https://ssrn.com/abstract=4810251 or http://dx.doi.org/10.2139/ssrn.4810251

Acharya, Kamal, Online Food Order System (May 2, 2024). Available at SSRN: https://ssrn.com/abstract=4814732 or http://dx.doi.org/10.2139/ssrn.4814732

Acharya, Kamal, University management system project. (May 1, 2024). Availableat SSRN: https://ssrn.com/abstract=4814103 or http://dx.doi.org/10.2139/ssrn.4814103

Acharya, Kamal, Online banking management system. (May 1, 2024). Available at SSRN: https://ssrn.com/abstract=4813597 or http://dx.doi.org/10.2139/ssrn.4813597

Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at SSRN: https://ssrn.com/abstract=4817534 or http://dx.doi.org/10.2139/ssrn.4817534

Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819626 or http://dx.doi.org/10.2139/ssrn.4819626

Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819630 or http://dx.doi.org/10.2139/ssrn.4819630

CHAPTER 8

APPENDICES

8.1 – SOURCE CODE

8.2 _O/P SCREENS